

Using the Secure Unattended Proxy (SUP)

Category: File Transfers

The Secure Unattended Proxy (SUP) allows users to perform remote operations on specific hosts within the HEC enclave (currently the Columbia front-ends, Pleiades front-ends/bridge nodes, Lou[1-2], and Susan)

without

the use of SecurID at the time of the operation. Users must obtain special "SUP keys" using SecurID authentication, after which those keys can be used to perform operations from unattended jobs and/or scripts.

SUP keys are currently allowed to call *scp*, *sftp*, *bbftp*, *qstat*, *rsync*, and *test*. In the future, other operations may be available via the SUP. Each SUP key is valid for a period of **one week** from the time it is generated. Users may have multiple SUP keys at the same time, which will expire asynchronously.

SUP Usage Summary

The steps below demonstrate how to quickly get up and running with the SUP using an scp transfer to pfe1 as an example. Consult the link in each step for full details (or simply read this page to completion).

1. Download and install client (one time)

```
your_localhost% wget -O sup http://hecc.nas.nasa.gov/kb/file/9
your_localhost% chmod 700 sup
your_localhost% mv sup ~/bin
```

2. Authorize host for SUP operations (one time per host)

```
your_localhost% ssh pfe1
pfe1% touch ~/.meshrc
```

3. Authorize directories for writes (one or more times per host)

```
your_localhost% ssh pfe1
pfe1% echo /tmp >> ~/.meshrc
```

4. Execute command (each time)

```
your_localhost% sup scp foobar pfe1:/tmp/c_foobar
```

5. Examine expected output (as needed)
6. Troubleshoot problems (as needed)

SUP Client

The SUP client performs all the steps needed to execute commands through the SUP as if the SUP itself did not exist. Commands that are allowed to pass through the SUP can be executed as if the remote host were directly connected by simply prepending the client command "sup". Besides executing remote commands, the client also includes an operating system-independent virtual file system that allows files across all SUP-connected resources to be accessed using standard filesystem commands.

• Requirements

The client requires Perl version 5.6.1 or above to execute and has been tested successfully on Linux, OS X, and Windows under Cygwin and coLinux. Only SSH is required to use the SUP, however, so if these requirements cannot be met, it is possible to use the SUP without the client.

Note for Windows users: even if the client is not used, scp and sftp require functionality only found in the OpenSSH versions of these commands, so Cygwin or coLinux will still be needed.

• Installation

1. Download the client and save to a file called "sup"
2. Make the client executable using "chmod 700 sup"
3. Move the client to a location in your \$PATH

• SSH Configuration

If your local username differs from your NAS username, it is recommended that you add the following to your `~/.ssh/config` file, where "nas_username" should be replaced with your NAS username:

```
Host sup.nas.nasa.gov sup-key.nas.nasa.gov
  User nas_username
```

NOTE: If you are using a config file based on the NAS config template, you do not have to do this step.

Alternatively, the client's -u option can be used as described in the next section. If your local username is the same as your NAS username, no additional configuration or command-line options are required.

- **SUP Command-line Options**

- ◆ -b

- Disable user interaction for use within scripts. Note that the client will fail if any interaction is required - normally only needed when your SUP key has expired or is otherwise unavailable.

- ◆ -k

- By default, the client leaves any SSH agents started on your behalf running for future invocations after the client exits. This option forces spawned agents to be killed before exiting. Note that "-b" automatically implies "-k".

- ◆ -u user

- Specify NAS username. Note that this option is required if your local username differs from your NAS username and you have not modified your SSH configuration appropriately.

- ◆ -v

- Enable verbose output for debugging purposes.

SUP Authorizations

The basic set of operations that may be performed using the SUP is specified by the administrator. To protect accounts from malicious use of SUP keys, users must grant execute and write permissions to SUP operations on each target system.

1. Execute Authorization

By default, even SUP operations permitted by site policy are not allowed to execute on a given host. To enable SUP operations to a given host (currently, the Columbia front-ends, Pleiades front-ends/bridge nodes, Lou[1-2], or Susan), the file `~/.meshrc` must exist on that host, which can be created by invoking the following:

```
touch ~/.meshrc
```

Note that the Pleiades front-ends/bridge nodes share their home filesystems, so this must only be done on one of these nodes. Similarly, the Columbia front-ends share

their home filesystems and the `~/.meshrc` file only needs to be created on one of the Columbia front-end nodes. Other systems must be authorized separately. Once this file exists on a host, all operations permitted by site policy are allowed to execute on that host.

2. Write Authorization

By default, SUP operations are not allowed to write to the file system on a given host. To enable writes to a given directory on a given host, that directory must be added (on a separate line) to the `~/.meshrc` file on that host. For example, the following lines in `~/.meshrc` indicate that writes should be permitted to `/nobackupp40` and `/tmp`.

```
/nobackupp40
/tmp
```

Each directory is the root of allowed writes, so this configuration would allow writes to all files and directories rooted at `/nobackupp40` and `/tmp` (for example, `/nobackupp40/some/dir`, `/tmp/some/file`).

Note that the root directory cannot be authorized. Also note that dot files (i.e. `~/.*`) in your home directory are never writable regardless of the contents of `~/.meshrc`.

Executing Commands Through SUP

Usage example of each command that may be executed through the SUP are given below. Note that SUP commands must be authorized for execution on each target host, and that transfers to a given host must be authorized for writes. Before a given operation is performed, the client may ask for certain information, including the existing or new passphrase for `~/.ssh/id_rsa`, the password + passcode for `sup.nas.nasa.gov`, and/or the password + passcode for `sup-key.nas.nasa.gov`.

File Transfer Commands

bbftp ([man page](#))

```
your_localhost% sup bbftp -e "put foobar /tmp/c_foobar"
pfe1.nas.nasa.gov
```

Note that you must use the fully qualified domain name of the target host (in this case, `pfe1.nas.nasa.gov`) if you are not within the NAS domain.

bbscp ([man page](#))

```
your_localhost% sup bbscp foobar pfe1.nas.nasa.gov:/tmp/c_foobar
```

Note that bbscp is just a client-side wrapper for bbftp, therefore, as with bbftp, you must use the fully qualified domain name of the target host (in this case, pfe1.nas.nasa.gov) if you are not within the NAS domain.

rsync ([man page](#))

```
your_localhost% sup rsync foobar pfe1:/tmp/c_foobar
```

If you intend to transfer files to your home directory, note that even if your home directory has been [authorized for writes](#), **rsync transfers to your home directory will fail unless the "-T" or "--temp-dir" option is specified**. This is because rsync uses temporary files starting with "." during transfers, which cannot be written in your home directory. You can avoid this problem by specifying an alternate temporary directory that is [authorized for writes](#). For example, the following example uses /tmp as the temporary directory when files are transferred to the home directory. Make sure that the temporary directory specified has enough space for the files being transferred.

```
your_localhost% sup rsync -T /tmp foobar pfe1:
```

scp ([man page](#))

```
your_localhost% sup scp foobar pfe1:/tmp/c_foobar
```

sftp ([man page](#))

```
your_localhost% sup sftp pfe1
```

File Monitoring Command

test ([man page](#))

```
your_localhost% sup ssh pfe1 test -f /tmp/c_foobar
```

Job Monitoring Command

qstat (man page available on Pleiades and Columbia)

```
your_localhost% sup ssh pfe1 qstat @pbspl1
```

SUP Expected Output

The following sequence shows the expected output for the command:

```
your_localhost% sup scp foobar pfe1:/tmp/c_foobar
```

for a user who has never used the SUP before.

The conditions under which each sub-sequence will be seen are indicated next to each header. Most of the items will only be seen once or during key generation. A second invocation will only show the command output portion.

1. Host key verification (seen once per client host)

```
No host key found for sup-key.nas.nasa.gov
...continue if fingerprint is
1b:9a:82:2b:b9:b0:7d:e5:08:50:1d:e8:14:76:a2:2e
The authenticity of host 'sup-key.nas.nasa.gov (129.99.242.7) '
can't be established.
RSA key fingerprint is
1b:9a:82:2b:b9:b0:7d:e5:08:50:1d:e8:14:76:a2:2e.
Are you sure you want to continue connecting (yes/no)? yes
No host key found for sup.nas.nasa.gov
...continue if fingerprint is
52:f3:61:9b:9c:73:79:4d:22:cb:f3:cd:9a:29:4e:fe
The authenticity of host 'sup.nas.nasa.gov (129.99.242.6) '
can't be established.
RSA key fingerprint is
52:f3:61:9b:9c:73:79:4d:22:cb:f3:cd:9a:29:4e:fe.
Are you sure you want to continue connecting (yes/no)? yes
```

2. Identity creation (seen during key generation if no identity available)

```
Cannot find identity /home/user/.ssh/id_rsa
...do you wish to generate it? (y/n) y
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
a3:cf:e5:50:12:6f:14:b1:21:59:19:a8:33:aa:77:40 user@host
```

3. Identity addition to agent (seen during key generation)

```
Adding identity /home/user/.ssh/id_rsa to agent
Enter passphrase for /home/user/.ssh/id_rsa:
Identity added: /home/user/.ssh/id_rsa
```

```
(/home/user/.ssh/id_rsa)
```

4. Identity initialization (seen once per identity)

```
Initializing identity on sup-key.nas.nasa.gov (provide login
information)
Password:
Enter PASSCODE:
Key a3:cf:e5:50:12:6f:14:b1:21:59:19:a8:33:aa:77:40 uploaded
successfully
```

5. SUP key generation (seen when no valid SUP keys available)

```
Generating key on sup.nas.nasa.gov (provide login information)
Password:
Enter PASSCODE:
```

6. Client upgrade (seen during key generation when new client available)

```
A newer version of the client is available (0.39 vs. 0.37)
...do you wish to replace the current version? (y/n) y
```

7. Command output (always seen)

```
foobar 100% 5 0.0KB/s 00:00
```

SUP Troubleshooting

The following error messages may be encountered during your SUP client usage. Note that the "-v" option can be given to the SUP client to output additional debugging information.

- "WARNING: Your password has expired"

This message indicates that your current password has expired and must be changed. To change your password, you must log in to an LDAP host (for example, Lou) through the SFEs and change your LDAP password. This change will be automatically propagated to the SUP within a few minutes.

- "Permission denied (~/.meshrc not found)"

This message indicates that you have not created a *.meshrc* file in your home directory on the target host. SUP commands must be authorized for execution on each target host.

- "Permission denied (unauthorized command)"

This message indicates that you have attempted an operation that is not currently authorized by the SUP. Check that the command line is valid and that the attempted command is one of the authorized commands. Certain options to authorized commands may also be disallowed, but these should never be needed in standard usage scenarios.

- Permission denied during file access (various forms)

These messages indicate that you attempted to read or write a file for which such access is not allowed. The most common cause is forgetting to authorize directories for writes. Reads and writes of ~/.* are never permitted.

Article ID: 145

Last updated: 02 Aug, 2011

Data Storage & Transfer -> File Transfers -> Using the Secure Unattended Proxy (SUP)

<http://www.nas.nasa.gov/hecc/support/kb/entry/145/?ajax=1>